

### Atividade Avaliativa

Leia atentamente cada questão.

Prazo de entrega: 18/06/2025

Parte 1 (SQL), você pode escolher resolver as questões em MySQL ou PostgreSQL.

Gere o script de criação do banco a seguir:

### Plataforma de Cursos Online

Você foi encarregado de desenvolver e gerenciar o banco de dados de uma nova plataforma de EAD. O modelo inicial é composto por Alunos, Cursos e as Inscrições que conectam ambos.



1.
  - a. Insira pelo menos 8 novos cursos na tabela curso.
  - b. Insira pelo menos 15 novos alunos na tabela aluno.
2.
  - a. Escreva um comando para atualizar o instrutor do curso com id = 1 para "Prof. Gladimir".
  - b. Selecione o nome dos alunos (nome\_aluno) e o nome dos cursos em que eles estão inscritos, mostrando apenas as inscrições da tabela inscricao com status 'ativa'.
3.
  - a. Crie um usuário chamado analista que tenha permissão **apenas de leitura (SELECT)** em todas as tabelas do banco aula14.
  - b. Crie um usuário chamado secretaria que possa **inserir, atualizar e consultar (INSERT, UPDATE, SELECT)** dados apenas na tabela inscricao.
4. Crie uma **PROCEDURE** chamada realizar\_inscricao(aluno\_id INT, curso\_id INT) que receba o ID de um aluno e de um curso e insira um novo registro na tabela inscricao.

5. Crie um **TRIGGER** chamado `log_nova_inscricao` que, após cada `INSERT` na tabela `inscricao`, insira um registro na tabela `log_inscricao`. O log deve conter o ID da nova inscrição (no campo `id_inscricao_ref`) e a ação "NOVA INSCRIÇÃO REALIZADA".
6. Crie uma **VIEW** chamada `v_inscricoes_detalhadas` que exiba o nome do aluno (`nome_aluno`), o email, o nome do curso e a data da inscrição.
7. Utilizando controle de transação (`START TRANSACTION`, `COMMIT`, `ROLLBACK`), execute os seguintes passos:
  - a. Inicie uma transação.
  - b. Insira uma nova inscrição para o aluno de `id = 2` no curso de `id = 3`.
  - c. Atualize o status dessa mesma inscrição para 'concluída'.
  - d. Se ambos os comandos forem bem-sucedidos, confirme a transação. Caso contrário, reverta todas as alterações.
8. (GROUP BY): Crie uma consulta que conte quantas inscrições cada curso possui. O resultado deve exibir o nome do curso e o total de alunos inscritos, ordenados do curso com mais alunos para o com menos.
9. Crie uma consulta que liste todos os cursos e a quantidade de alunos inscritos em cada um.  
A consulta deve incluir também os cursos que não possuem nenhum aluno inscrito (mostrando o valor '0' como total).  
Utilize `LEFT JOIN` para garantir que todos os cursos da tabela `curso` apareçam no resultado.
10. Crie uma consulta que funcione como um relatório de inscrições por status. O resultado deve ter uma linha para cada nome de curso e duas colunas de contagem: uma chamada `total_ativas` e outra `total_concluidas`.  
Utilize a técnica de pivoteamento com `SUM` e `CASE` para gerar este relatório.

"Exemplo" de resultado esperado:

nome_curso	tot_ativas	tot_concluidas
Banco de Dados para Big Data	1	0
Cibersegurança Essencial	1	0
Engenharia de Software	1	0
Gestão de Projetos Ágeis	0	1
Inteligência Artificial	1	0
Introdução a Algoritmos	1	0
Machine Learning	1	0

**Parte 2 (NoSQL), utilize o MongoDB.**

Crie um novo banco de dados chamado `plataforma_ead_nosql`.

**11.**

a. Crie uma coleção chamada `inscricoes`.

b. Para que seja possível testar todos os comandos a seguir, insira pelo menos 15 documentos nesta coleção.

Varie os dados para que existam nomes de cursos, nomes de alunos e datas diferentes.

Nem todos os documentos não precisam ter os mesmos campos.

Exemplos:

```
{
  "aluno": { "nome": "Carlos Andrade", "email": "carlos.a@email.com" },
  "curso": { "nome": "Análise de Dados com Python", "instrutor": "Prof. Silva" },
  "data_inscricao": new Date("2024-10-25"),
  "status": "ativa"
},
{
  "aluno": { "nome": "Mariana Costa", "email": "mari.c@email.com" },
  "curso": { "nome": "Banco de Dados para Big Data", "instrutor": "Prof. Silva" },
  "data_inscricao": new Date("2024-11-05"),
  "status": "ativa"
},
{
  "aluno": { "nome": "Ana Beatriz", "email": "ana.b@email.com" },
  "curso": { "nome": "Introdução a Algoritmos", "instrutor": "Prof. Souza" },
  "data_inscricao": new Date("2025-02-15"),
  "status": "ativa"
},
{
  "aluno": { "nome": "Pedro Martins", "email": "pedro.m@email.com" },
  "curso": { "nome": "Machine Learning", "instrutor": "Prof. Souza" },
  "data_inscricao": new Date("2025-03-01"),
  "status": "ativa"
},
{
  "aluno": { "nome": "Juliana Lima", "email": "ju.lima@email.com" },
  "curso": { "nome": "Desenvolvimento Web Fullstack", "instrutor": "Prof. Gladimir" },
  "data_inscricao": new Date("2025-01-20"),
  "status": "concluída"
},
{
  "aluno": { "nome": "Amanda Gomes", "email": "amanda.g@email.com" },
  "curso": { "nome": "Engenharia de Software", "instrutor": "Prof. Carla" },
  "data_inscricao": new Date("2025-04-10"),
  "status": "ativa"
},
{
  "aluno": { "nome": "Lucas Pereira", "email": "lucas.p@email.com", "matricula": "BR25001" },
  "curso": { "nome": "Cibersegurança Essencial", "instrutor": "Prof. Carla" },
  "data_inscricao": new Date("2025-03-20"),
  "status": "ativa",
  "bolsa": { "tipo": "Mérito Acadêmico", "percentual": 100 }
},
```

```
{
  "aluno": { "nome": "Fernanda Dias", "email": "fernanda.d@email.com" },
  "curso": { "nome": "Gestão de Projetos Ágeis", "instrutor": "Prof. Gladimir" },
  "data_inscricao": new Date("2025-05-01"),
  "status": "concluída",
  "nota_final": 9.5,
  "empresa_contratante": "Tech Solutions Inc."
},

{
  "aluno": { "nome": "Ricardo Neves" },
  "curso": { "nome": "Tópicos Avançados em Banco de Dados" },
  "data_inscricao": new Date("2025-05-10"),
  "tags": ["SQL", "NoSQL", "Performance"]
},

{
  "aluno": { "nome": "Vitor Hugo", "email": "vitor.h@email.com" },
  "curso": { "nome": "Lógica de Programação (Legacy)", "instrutor": "Prof. Antigo" },
  "data_inscricao": new Date("2024-03-15"),
  "status": "arquivada"
}
```

### 12. (expressão regular)

- Escreva uma consulta para encontrar todas as inscrições em cursos que contenham a palavra **"Dados"** em seu nome.
- Escreva uma consulta para encontrar todas as inscrições de alunos cujo nome comece com a letra **"A"**.

### 13. (updateMany)

O "Prof. Souza" mudou-se para outra instituição.

Escreva um comando que atualize **todas** as inscrições dos cursos ministrados por ele, alterando o campo status para **"em espera"**.

### 14. (replaceMany)

A plataforma decidiu arquivar todas as inscrições feitas pelo instrutor "Prof. Silva".

Escreva um comando que **substitua o documento inteiro** de todas as inscrições dele por um novo documento com a seguinte estrutura: { "inscricao\_arquivada": true, "aviso": "Curso migrado para sistema legado." }.

**15. (deleteMany):** Escreva um comando que remova **todas** as inscrições com o status **"em espera"**.

**16. (aggregate):** Conte quantas inscrições existem para cada instrutor.

O resultado deve mostrar o nome do instrutor e o total de inscritos.

**17. (createIndex):** Crie um índice no campo aluno.email da coleção inscricoes para otimizar futuras buscas por email de alunos.

**Obs.:** As respostas deverão ser enviadas em um único arquivo. Preferencialmente texto (.txt) ou PDF (.pdf)